

TINA - Knowledgebased Mission Planning for Future Spacecrafts and their Autonomous Operation

Jens Eickhoff, Harald Eisenmann, Oliver Kienzler

Dornier Satellitensysteme GmbH,

P.O.Box 1420, D-88039 Friedrichshafen

eickhoff@spacediv.dofn.de

Abstract

The TINA Systems (TImeliNe Assistants) are a family of knowledgebased planning systems developed by Dornier Satellitensysteme GmbH. The first approaches to knowledgebased planning systems reach back to the eighties and cover the domain of satellite assembly planning, using rule based and Lisp based problem solving techniques.

Over the years, these planning tools have evolved to a modular object oriented program library with the application domains of satellite assembly, integration and test as well as satellite mission operations. One common kernel includes the timeline generation algorithms and the inference and constraint propagation engine. A commercially available constraint propagation toolset is used as base and the entire application is implemented in the C++ language. Coupled hereto are a graphical X-Windows based user interface and domain specific numeric modules, such as e.g. an orbit propagation module for the mission planning application.

The key feature, which makes the TINA Timeline Generator specially performant in complex problem solving domains, is the fact that it is neither a pure planning algorithm for implementation of logical PERT charts, nor a pure Scheduler (like a lot of PC based tools) for generating GANTT Timelines. TINA is a "Timeline Generator", a tool considering in an integrated timeline processing procedure both logical interdependencies of positioned activities as well as numeric constraints such as resource consumptions or interrelations of generated and consumed resources.

This hybrid base technology also made TINA the appropriate choice for a recently completed study of the European Space Agency (ESA) which assessed the applicability of autonomous operation of earth observation satellites. Earth observation satellites today still require a full 24hr operated ground station and satellites with a near polar, highly inclined orbit additionally impose the complications that not at every orbit a ground station contact with mission product downlink and command uplink can be achieved. Severe operational inefficiencies occur, when during such an off-contact period errors occur during timeline execution onboard the satellite.

To improve mission product availability and for assessment of a satellite operation with a ground station operated only during normal business hours, within this ESA study "Distributed Intelligence for Ground/Space Systems" an intelligent mission timeline generation, timeline uplink at ground contact and timeline regeneration onboard the satellite in case of problems has been assessed. Within this study an entirely new commanding concept for satellite payloads based on parameterized "user requests" has been developed and for the first time has been implemented in a timeline generation system.

1. Introduction

The TINA Systems (TImeliNe Assistants) are a family of knowledgebased planning systems developed by Dornier Satellitensysteme GmbH. The first approaches to knowledgebased planning systems reach back to the eighties and cover the domain of satellite assembly planning, using rule based and Lisp based problem solving techniques. Today's descendants of the TINA planning system

family are based on a modular C++ software architecture featuring an entirely new algorithmic concept.

This new type of planning systems focuses on problem domains, where the logic of the plan (sequential arrangement of activities on the timeline) itself is fundamentally influenced by time and resource constraints. Thus TINA focuses on problem domains, where an integrated approach to consistent solving of the logic of the activities (planning) and a matching of all time and resource constraints (scheduling) is required. Due to this hybrid technique, TINA is called a "Timeline Generator" in contrast to a pure planner or scheduler.

1.1 Application Domain

One of most demanding planning/scheduling domains in satellite operations is the detailed payload operation planning for the upcoming new generation of autonomous earth observation satellites and deep space probes.

Former and current satellites due to required radiation resistance of the onboard computers still are equipped with CPU types like the 31750 or MC 68000 to 68020 (thus corresponding to the low-end PC class). Based on such computers, the commanding of satellites had to be based on a technique requiring the least possible computational performance during execution. Therefore today's satellites are purely based on time tagged commands. This means that every mode switch of any of the satellites payloads or subsystems to be performed onboard is coded as command and has an associated execution time stating, when it will be executed onboard.

These timelines of time tagged commands are generated on ground and are uplinked to the satellite each time it passes the visibility range of a ground station. For scientific earth observation satellites with mostly highly inclined (near polar) orbits, usually near polar ground stations are used and ground contact times typically lie in the range of one 5 minutes contact every 3 to 7 orbits.

If now during execution of such a time tagged command set any failure occurs - most of them not being really severe equipment damages - the affected payload goes into a refuse mode, performing no further operations until being reset from ground during next ground contact. To avoid mission product loss for several orbits due to trivial causes and to provide more flexibility in timeline execution onboard new solutions are coming up the horizon.

With the space qualified implementation of a SPARC CPU.- the european ERC32 -, for the first time onboard computers reach a performance range which

opens up the possibility to use more intelligent spacecraft commanding techniques. Envisaged are:

- The generation of timelines based on the earth observations user requests on ground, uplink of the timeline and situation dependent adaption onboard
- The pure uplink of the user requests directly to the spacecraft, entire mission planning onboard and tracking of onboard activities by the ground station. A variant of this type of operational scenario would be a globe wide direct access of the users to the spacecraft via small decentral satellite transceivers, each user linking up his requests and receiving his mission product after processing.

For evaluation of the pro's and contra's of these commanding concepts and for evaluations of numerical performance aspects and traceability of the onboard processing from ground, several studies have been initiated by the european space agency and TINA has been selected as base technique (see e.g.[4]).

1.2 System Configuration

The TINA Timeline Generator is based on a modular program architecture, in its standalone version consisting of three separate processes interlinked via data connections:

- a Tcl/Tk based graphical user interface
- the TimelineGenerator kernel,
- an orbit propagator for computing orbital positions, target visibilities and orbital events (e.g. eclipse).

The TimelineGenerator is controlled by the graphical user interface and it can contact the orbit module to compute orbit information necessary within the timeline generation process.

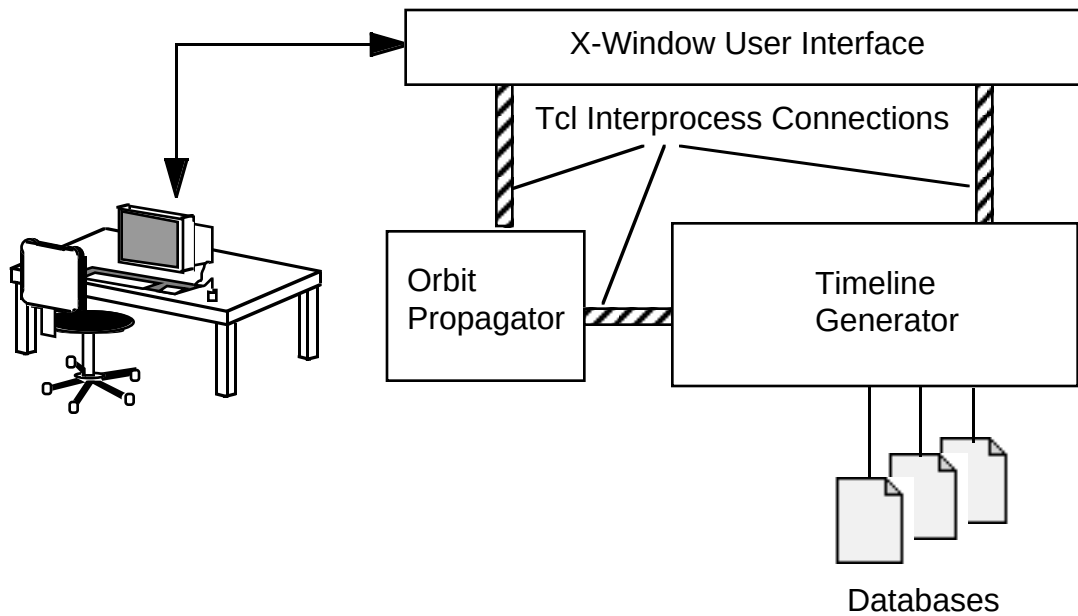


Figure 1: Process Architecture of the TINA Timeline Generator

2. Base Technology

2.1 Problem Formulation Techniques

The latest implementation, the TINA Timeline Generation library now is based on a pure C++ technology, using the commercial constraint propagation libraries ILOG Solver and ILOG Schedule from ILOG. S.A. France. The TINA Timeline Generation approach features an integrated approach for resolving logical and numeric interdependencies of timeline items in an integrated loop to avoid timeline deficiencies that cannot be solved in a sequential planning - scheduling approach.

The algorithm is designed so that most of its functionality is formulated in algorithmic manner. Special steps of the Timeline Generation process can be supported by planning knowledge formulated as rules interpreted by an inference processor (based on ILOG Rules). Thus a high performance of the overall planning process can be achieved.

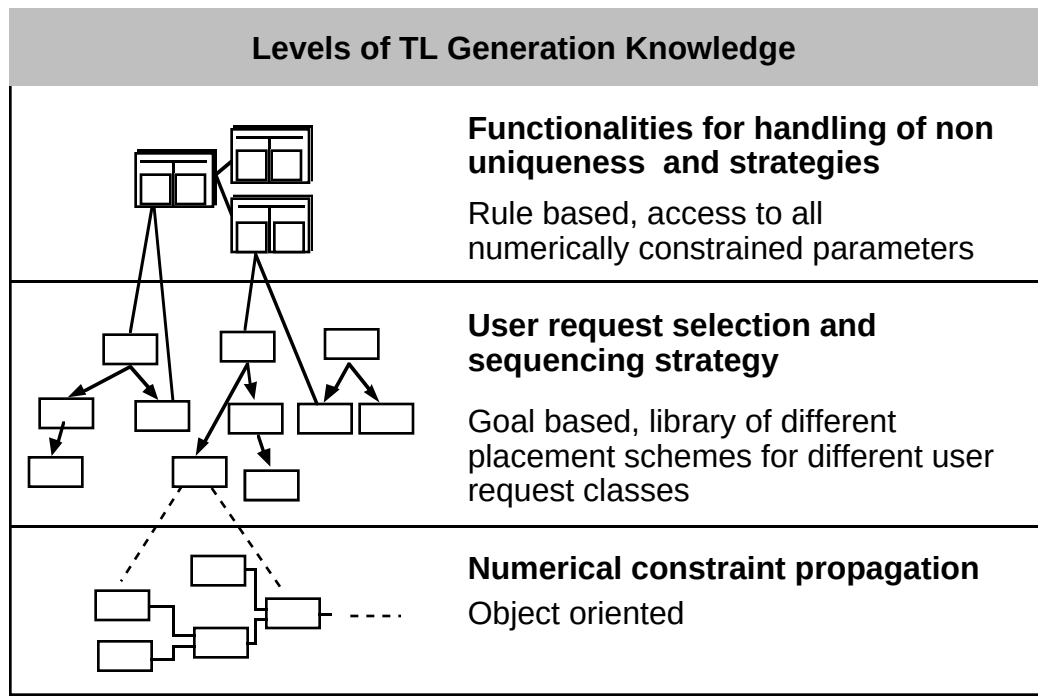


Figure 2: Hierarchy of Functionalities and their Implementation Techniques

The basis for the overall problem formulation is the object oriented modelling of the activities to be carried out onboard the satellite, the available and consumed resources and operational constraints. During operation of the timeline generation process, at each attempt to place an activity on the timeline, the whole network of constraints is checked and modifications instantly are propagated so that no single constraint is violated. The know how of descriptive problem formulation is implemented using this functionality.

The key functionality of the timeline generation process however is the goal based activity selection and placement functionality. This goal based algorithm keeps track of all the changes it applies subsequently and if an activity placement decision turns out to fail instantly or even after several processing steps, it can retract the activity from the timeline and return to the conditions before its placement. The algorithmic know how about time efficient and flexible timeline generation is implemented here.

Finally for the implementation of placement strategies which have to be editable by the user without recompiling the program, an interface to a rule chaining system is provided. Such strategies e.g. can be useful for for special cases of non uniquely decideable placement problems or mission dependent preferences of certain satellite users.

2.2 Domain Modelling

The following paragraphs will explain the descriptive functionalities used to model the timeline generation problem. These are User Requests, Payload- and Equipment Queues, Activities and Resources.

2.2.1. User requests

User Requests are the input to the Timeline Generator during operation. A user requests software item corresponds to a specification set, a satellite payload user submits to the space operations center to request e.g. an earth observation data set. Such a user request specification includes e.g. the coordinates (from/to) which shall be observed, a time frame when the observation has to be started earliest and completed latest, the payload to be used for the observation (e.g. an interferometer), payload settings (e.g. optical filters to use) etc.

All these data are included in a user request database. This database is read at start of a TINA run and during timeline generation all necessary activities to fulfill a user request are placed on the timeline, considering initial conditions, orbital configurations, resource constraints etc.

Each user request has a so called **key activity**. This normally is the main operational activity to be carried out for a user request, e.g. the satellite payload operation. Certain preparatory activities are required for setting up the payload for observation, such as orbital manoeuvres or instrument adjustment, focusing, warming etc. and other post-operational activities are required to shutdown again the payload to a non operational state. The relations to the key activity of these preparatory and post-operational **support activities** are defined in a mission specific user request type database (see also fig. 4).

2.2.2. Payload- and Equipment Queues

They are descriptive objects used to model satellite payloads and key equipment. User requests of different type may utilize the same payload. Queues are therefore used to model non-parallelities in the placement of user requests on a timeline. E.g. the calibration of a payload and an earth observation with the same payload cannot take place at the same time. Therefore the calibration user requests and the operation user request type for a payload are grouped within one payload queue.

2.2.3. Activities

The activities of a satellite mission are the central items to be scheduled. In the TINA systems the placement of an activity is characterized by

- initial constraints that are necessary to start the activity
- resources that have to be available to carry out the activity

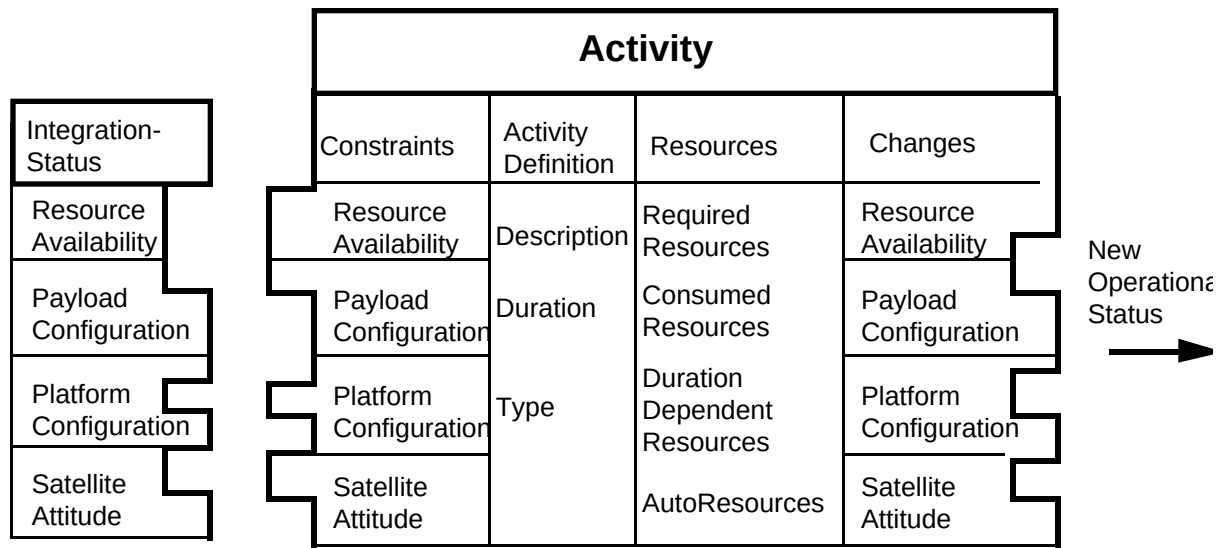


Figure 3: An Activity in the TINA Timeline Generation Process

An activity itself is characterized by changes in the configuration status it applies. In TINA the new status after an activity has been carried out is described by the changes the activity applies to its initial integration status and the resources it consumes. The complete mission status is automatically propagated from the initial status at the beginning of the timeline and the summation of all status changes resulting from all activities up to the actual position. TINA distinguishes between three types of activities:

- key activities,
- state activities
- switch activities.

The key activity of a user requests serves for all operations in conjunction with the marking of a user request as placed/non-placed, its orbit constraints etc. The switch activities serve for modelling the logical transitions of modes in switch-up and -down of equipment and the state activities serve for modelling the resource consumption of equipment when being set to a certain operational mode (e.g. power consumption, memory tape storage consumption etc.).

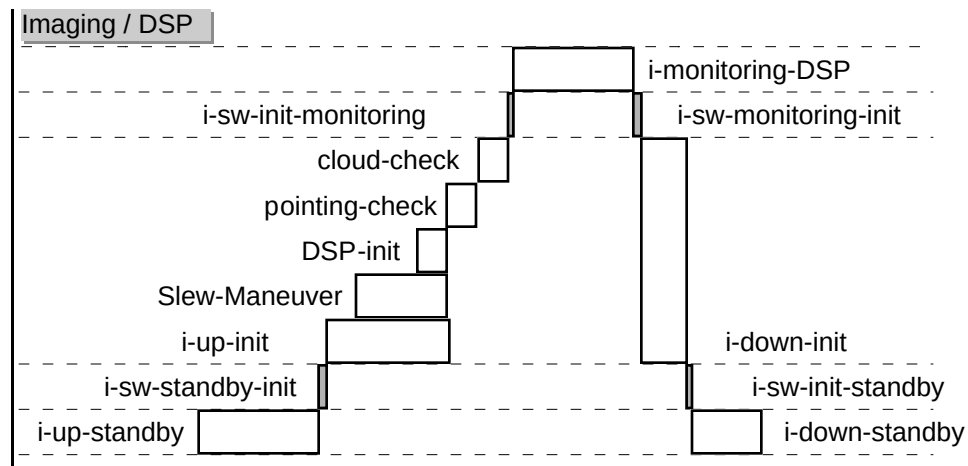


Figure 4: Switch and State Activities for an Imaging User Request

A special variant of activities represent the so called **AutoActivities**. These types of activities are used to model automatically occurring resource generation or consumption according to orbital position of the satellite in eclipse or in sunlight, e.g. the generation of electric energy by the solar panels of a satellite or the evaporation of cryocoolant of infrared instruments. Mode details on these activities are provided in the following chapter as they are directly concerned with resources.

2.2.4. Resources

To position an activity on the timeline certain resources have to be available (e.g. electric energy). Such resources can be limited concerning temporal availability and quantity. TINA distinguishes between

- consumed resources whereof a special variant are the duration dependent resources (e.g. electrical energy in [Ah]),
- required resources which are available again after completion of the activity (e.g. a telecommunication channel occupied by a data downlink activity, but freed again after activity completion).

Resources of the "consumed" type of course also can be produced. The electric energy stored in the satellite's batteries for example is generated by the solar panels when the satellite is in the sunlight phase of its orbit.

To model orbit related automatisms in resource management a special functionality is implemented in TINA - the already mentioned AutoActivities. By appropriate definition in a specific input database, the so called **autoactivity.db**

the TINA user can specify the dependency of orbit based automatic production or consumption of a resource. According to the specified input TINA places AutoActivities on the generated timeline, which constantly generate (respectively consume) the specified resource during orbit sun phase or eclipse respectively.

Application examples is the automatically generated electric energy supplied by the solar panels as soon as the satellite leaves the eclipse or thermostat controlled automatic heating of certain satellite components as soon as the satellite enters the eclipse and associated energy consumption.

3. Output of the Timeline Generation

The result output of a timeline generation run is stored in several files and can be visualized. The graphical output consists of

- an Orbit Plot showing the ground track of the satellite within the time period covered by the actual timeline (see fig. 5),
- the Timeline Gantt Chart reflecting the computed position of the activities on the timeline (see fig. 6) and
- the Resource Chart reflecting the resource budgets over time (see fig. 7).

At each end of a timeline generation run, TINA writes out the new initial state information, including resource budgets at timeline end and equipment switch states as well as an update of the user request database stating which user requests have been place in the current timeline and which still are open to be placed, as e.g. no target visibility was available so far. Thus TINA provides the possibility to compute several timelines for several orbits within one computation run (the state information at end of one timeline is taken as initial state for the generation of the following) and the corresponding visualization of the multiple generated timelines. The duration period of a timeline (usually one orbit) is selectable by the user.

4. TINA User Interface

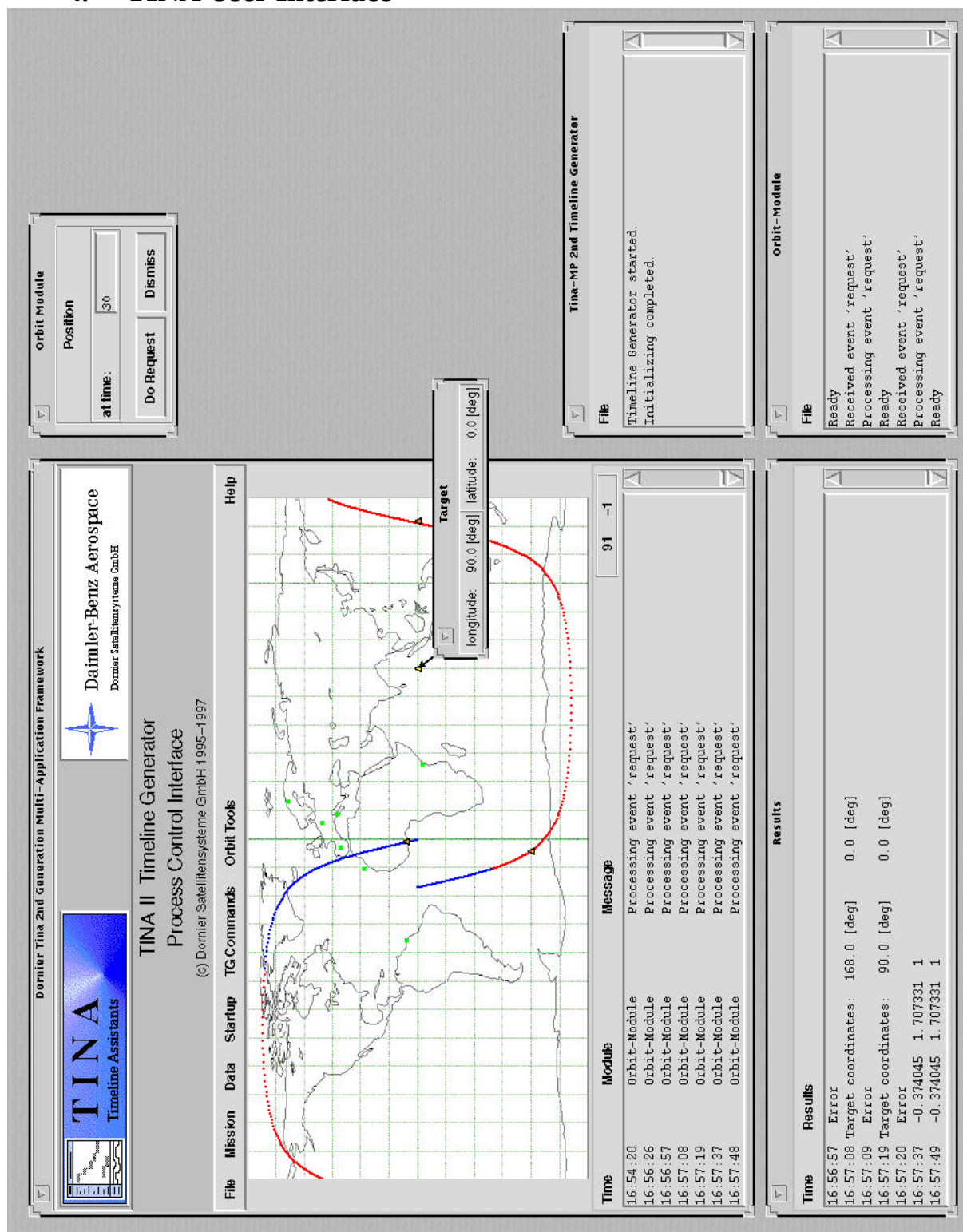


Figure 5: TINA User Interface with Orbit Module and Timeline Generator

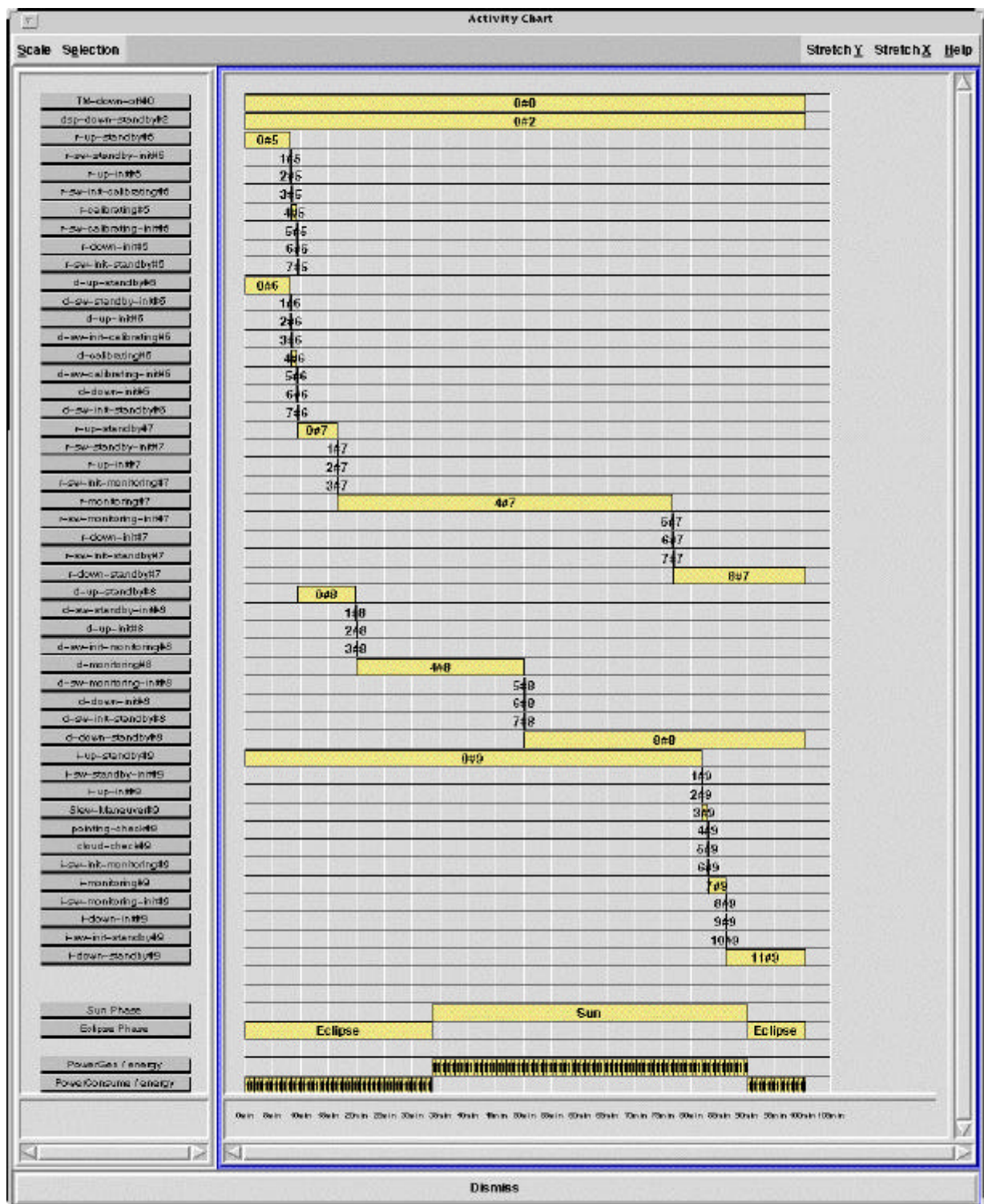


Figure 6: Activity Timeline

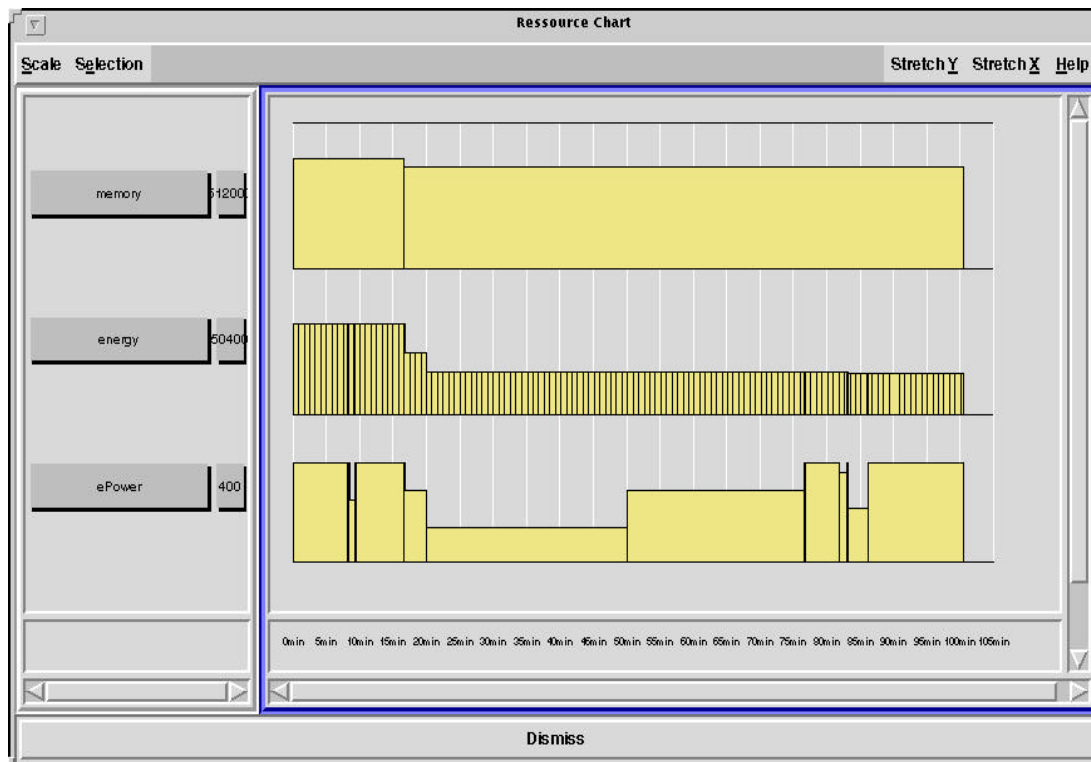


Figure 7: Resource Timeline

5. Preparatory Projects to Autonomous Spacecraft Operation

The most important application field for the TINA Timeline Generator is the preparatory work for operation of future, more autonomous satellites. The conventional operation concept of earth observation satellites is based on timelines with time tagged macrocommands corresponding to the detailed mode switches of payloads and platform components. The commanded units fall back to a safe or refuse mode of in case of the slightest deviations between its state and prescribed command entry state (e.g. a payload not preheated enough). Furthermore always the whole switch cycle of a macrocommand has to be carried out. It is not possible e.g. to perform the switch on sequence of an instrument, make an observation and then to keep the instrument in heated mode for a following observation just some minutes later.

More flexible commanding concepts focus on entire timeline generation onboard and uplink of the user requests only or on uplink of prepared timelines and adaptations and recovery timeline generation in orbit in case of problems. Some research studies of the European Space Agency ESA are focusing on these spacecraft autonomy technologies. The main requirement to an onboard autonomy system is the quick response to failures and thus a highly efficient timeline generation. The TINA Timeline Generator currently is used in

- an ESA study on system autonomy modelling a mock-up of a combined ground/space system simulating scenarios for realistic modelling of satellite operations. Two UNIX workstations are used for modelling both the satellite (running a satellite simulator) and the ground station [3,4].
- an ESA study for evaluation of the numeric performance of the goal and constraint based numeric technique of timeline generation. The suitability of TINA for integration into a satellite onboard software is assessed here [5].

5.1 Distributed Ground/Space Architecture for Operation of Autonomous Satellites

Performing mission operations with optimized commanding techniques leads to scenarios, where the onboard resources of the satellite (such as e.g. power) will be used for mission product acquisition as far as possible. Assuming a ground based mission timeline generation and spacecraft monitoring the important aspects to be considered are the possible deviations between the theoretically computed

resource budgets taken as basis for timeline generation on ground and the actual values available on board.

Without constant synchronization of the values at each ground contact, deviations between actual onboard values and theoretic values used by the ground system would accumulate over time. This can lead to severe failures onboard the spacecraft and loss of mission product. To avoid the accumulation of such deviations it is necessary to adjust the resource profile held in the ground segment to the actual value downlinked from the satellite at each ground contact time.

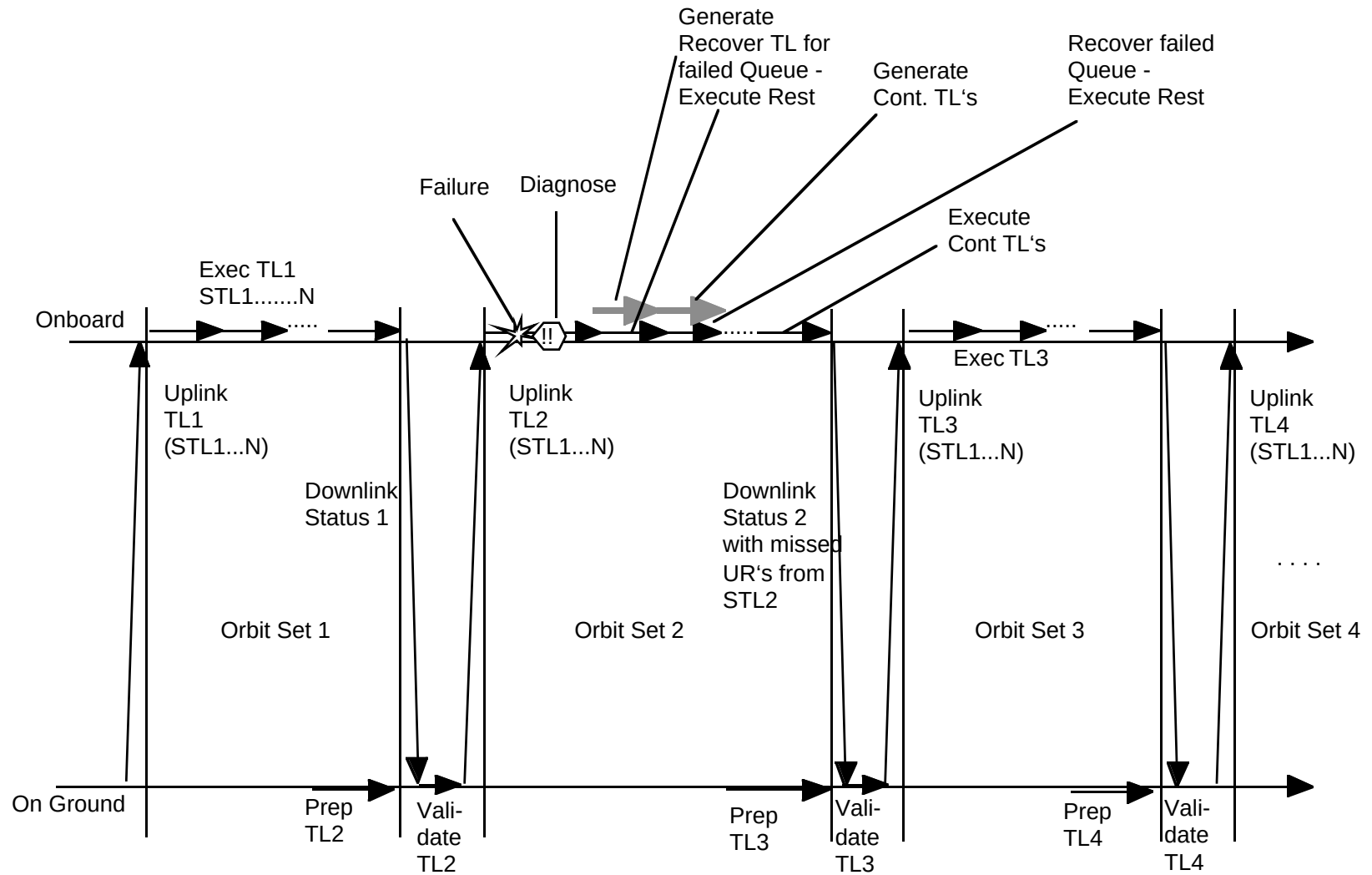
Even greater deviations in the resource profiles and additionally in the user request databases might occur, if failure recovery operations are performed autonomously onboard the spacecraft.

Figure 8 shows an assessed failure recovery onboard the spacecraft taken from [3] as example. The scenario serves for demonstration of the update strategy between space- and ground segment and the strategy to consider the changed onboard status in future timelines.

For appropriate reaction to onboard failures (both resource and unit failures) a handling of the timeline which covers an entire orbit-set (the amount of orbits from ground coverage to ground coverage) is not appropriate. Instead of an entire timeline it is necessary to split the timeline into a set of subtimelines (STL1 to STL N in figure 8) each of them covering a subset e.g. one orbit).

Figure 8:

Synchronization between S/C Status and Ground System.



5.2 Performance Assessment

An ongoing ESA software assessment project serves to identify the feasibility of the constraint based mission timeline generation technology of TINA for onboard activity scheduling in autonomous spacecrafts. Particularly numerical characteristics and memory requirements of the current TINA C++ architecture are being assessed, also considering the future integration into the hard-realtime software architecture of a satellite ADA onboard environment. The scenarios used within this study model the operation of a small scientific satellite with a highly inclined near polar orbit. In detail the resources and activities for operation of

- the satellite platform
- an optical imager payload,
- a radiation detection payload and
- a space debris detection payload
- the telecommanding and telemetry equipment

are considered.

6. Ongoing Development

Future development will focus on the setup of an architecture which integrates the C++ Timeline Generator into an Ada 95 frame architecture making the Timeline Generator running under Ada memory management and tasking control. Based on this setup the detailed memory and processing resource parameters for an onboard timeline application of TINA in an autonomous satellite will be assessed.

Keywords

TINA, Knowledgebased Planning Systems, Constraint Propagation, Timeline Generation, Satellite Mission Planning, .Autonomous Satellite Operations

Literature

- [1] Gautier G.:
*TINA - Timeline Assistant for Planning of Spacecraft Assembly,
Integration and Verification*

Workshop on Artificial Intelligence and Knowledge-Based Systems for
Space, ESA/ESTEC Noordwijk, Holland, May 22 nd - 24 th 1991

- [2] Eickhoff, J.; Hendricks, R.; Urban, F.; Eisenmann, H.:
*TINA - A Knowledgebased, Modular Planning System for Satellite AIT,
Crew Training and Mission Support*
4th Workshop on Artificial Intelligence and Knowledgebased Systems for
Space, ESA/ESTEC, Noordwijk, Holland, 17. - 19. May 1993

- [3] N.N.:
Mission Scenario Definition, Phase 2,
Distributed Intelligence for Ground/Space Systems,
Dornier Satellitensysteme GmbH, DI-DOR-TN-003, Issue 1, Dec. 1996

- [4] Aarup, Mads; Fuchs, Joachim; Eickhoff, Jens; Khan, Zahoor:
Distributed Intelligence For Ground/Space Systems,
6th Workshop on Artificial Intelligence and Knowledgebased Systems for
Space, ESA/ESTEC, Noordwijk, Holland, October 1995

- [5] Eickhoff J.:
TN on Mission Scenario Definition,
Scheduler Requirements and Performance Assessment,
Dornier Satellitensysteme GmbH, SCH-A-TN-001, Issue 1.1, Feb. 1997